Defense Information Infrastructure (DII)

Common Operating Environment (COE)

Windows NT 4.0

Programmer's Reference Manual

Version 3.1

2 April 1997

Prepared for:

Defense Information Systems Agency

Table of Contents

Table of Contents **(continued)**

This page intentionally left blank.

Foreword

Details regarding the Defense Information Infrastructure (DII) Common Operating Environment (COE) Commercial-Off-The-Shelf (COTS) licensed products do not appear in this release of the DII COE Programmer's Reference Manual. Contact the Defense Information Systems Agency (DISA) for information concerning the licensed software and supporting documentation.

> **NOTE**: Contact DISA's Configuration Management for the available documentation on the DII COE COTS licensed applications.

The DII COE Reference Programmer's Manual is part of a set of DII COE developer documentation published in conjunction with major or general releases of the DII COE (such as DII COE Version 3.0). It contains the latest information available on the date of release of this publication.

Segment Availability: Not all DII COE segments described in this document may be available in the referenced release of the DII COE. The information contained herein may precede the availability of certain DII COE software segments. Please visit the DISA's DII COE Home Page on the World Wide Web to obtain the most current information available and to obtain the latest available release of DII COE software segments and other related documentation required by your organization.

DISA DII COE Home Page URL: http://www.disa.mil/dii/diicoe or http://204.34.175.79/dii/

Segment Documentation Applicability: Software segments included in this document are identified by their DII COE segment release version number and/or product version number, as available. It is important to note, however, that documentation released with a given segment version may be applicable to subsequent version(s) of the same software segment. For example, the installation guide for a DII segment version 3.0.0.3 may also apply to version 3.0.0.4 of the same segment unless superseded by a new release of the documentation. Refer to the details provided in the Version Description Document (VDD) for a particular segment release and its related amendments or errata to obtain the most current information on the fixes incorporated, additional sources of information, or reference documents.

This page intentionally left blank.

1. Introduction

1.1 Purpose

The *DII COE Programmer's Reference Manual* is intended to be a reference document for information about the Defense Information Infrastructure (DII) Common Operating Environment (COE). This manual has been created, in conjunction with the *DII COE Programmer's Manual*, to support programmers' development activity in the DII COE.

This manual provides specific information concerning the component structure and toolsets currently being used in the DII COE, and serves as a catalogue of published standards and COE related documentation. Part of the concept of this manual is to provide new COE users and programmers with a quick reference guide to applicable standards and specifications, including Application Programmer's Interface (API) documents, which control and shape system and application development within the DII COE architecture.

The contents of this document apply only to Intel compatible personal computer (PC) workstations running Microsoft Windows New Technology (NT) 4.0 operating system.

The Defense Information Systems Agency (DISA) will review and update this document as required to remain current with the evolution of the DII COE. This document supersedes Version 2.0, all earlier draft versions, presentations, or working group notes. Please direct any comments or questions regarding the DII COE Programmer's Reference Manual to:

Point of Contact (POC)	Cmdr. Charles B. Cameron Chief, DII COE Engineering Division
Address	DISA/JIEO/JEXF-OSF 45335 Vintage Park Plaza Sterling, VA 20166-6701
Telephone	(703) 735-8825
Fax	(703) 735-8761
Internet Address	cameronc@ncr.disa.mil

1.2 Scope

This document describes the technical aspects of the DII COE, its major components, and the resources provided for COE development activities. It describes the various segments for the Sun workstation, and the purpose and usage of each segment.

This document provides implementation details which describe, from the perspective of DII COE software development, the following:

- the standards and reference documents which govern the DII COE,

- the COE development process,
- the on-line resources which support COE development,
- the guidelines for COE development,
- the COE developers' toolkit,
- the Application Program Interfaces (APIs) supported in the current COE release, and
- the components of the current COE release.

1.3 DII COE Overview

The DII COE is best described as an architecture that is fully compliant with the *Department of Defense, Technical Architecture Framework for Information Management* (TAFIM). The COE is a foundation for building an open system designed around a client/broker/server model. The DII COE encompasses architecture, standards, reusable software modules, and automated integration tools in a cohesive framework for systems development.

Each system built upon the COE foundation uses the same set of APIs to access common COE components, the same approach to integration, and the same set of tools for enforcing COE principles. Precisely the same COE software components are used for common functions, such as communications interfaces and dataflow management.

1.4 COE Tools Overview

The COE tools were developed to perform selected DII COE tasks. The tools may be executed from the command line, menus, or a code segment using published APIs. Detail information on the use of the COE tools and how they communicate with the outside world are available from the *DII COE Integration and Runtime Specifications* (DII COE I&RTS). A list of the COE tools available that help developers create, install, and test segments is provided in the DII COE Programmer's Reference Manual. In all cases, the final authority on the use of a tool is the associated API reference guide.

1.5 API References

The DII COE, by its very nature and design, is transitional and evolutionary. Information available concerning the DII COE APIs can be found in the DII COE documentation listed in an Appendix of the DII COE Programmer's Reference Manual. For detailed information on APIs, refer to the specific API guide.

1.6 Document Structure

This document is divided into 3 major sections.

Section 1 discusses the purpose and scope of this document. It also contains an overview of the DII COE.

Section 2 discusses standards and reference documentation.

Section 3 describes DII COE components and segments of Windows NT 4.0.

Appendices There are three appendices (A through C) as follows:

A. Acronym List
B. List of COE Tools & APIs (Windows NT 4.0)
C. List of POSIX Function Calls
D. Application Programmer Interface (API) Reference Guide for Windows NT 4.0.

This page intentionally left blank

2. Standards and Reference Documents

This section will cover the COE environment in terms of applicable standards. Appendix E of the *DII COE Programmer's Manual* covers the list of mandated technolgies referenced by Joint Technical Architecture (JTA) which are derived from the recommended standards listed in the Technical Architecture Framework for Information Management (TAFIM). The TAFIM list the Adopted Information Technology Standards of the TAFIM.

The DII COE reflects the principles, architecture, and standards adopted by TAFIM which includes the Portable Common Toolkit Environment (PCTE) and the ongoing work in the Integrated Software Engineering Environment (ISEE).

2.1 Reference Documentation

Refer to Appendix C of the DII COE Programmer's Manual for a complete list of reference documents.

POSIX.1 is the Portable Operating System Interface that is currently a Federal Information Processing Standard (FIPS) 151-2 based on ISO/IEC 9945-1 international standards and derived from IEEE POSC P1003.1 American National Standards Institute (ANSI) Standard. POSIX.2 is part 2 of the Shell and Utilities and is currently the FIPS 189 standard based on the ISO/IEC 9945-2 international standard and derived from IEEE POSC P1003.2 (ANSI) Standard.

2.2 List of POSIX Calls

POSIX, which originated in the Institute of Electrical and Electronics Engineering (IEEE) standards committee for the Portable Application Standards Committee (PASC), refers to a Portable System Interface for compliant computer environments.

There are three parts to the implementation of the standard:

- Definition of terms, global names and concepts,
- Interface facilities, and
- Data interchange format.

A list of POSIX function calls (APIs) appears in Appendix C. They are part of the interface and are required for use in the programming environment to effect portability which assures that programs can interface reliably with any POSIX-compliant operating system using the calls' specifications.

2.3 Standards References

The framework for major service areas of the TAFIM applies to the DII COE. Conversely, the DII COE is compliant with the TAFIM's classification of standard services as well as the requirements for JTA mandated technology standards.

The Computer-Aided Software Engineering (CASE) tools and environments specific to COE tools are compliant to the principles of the Portable Common Toolkit Environment (PCTE). PCTE is the ISO/IEC 13719 Standard used with C and Ada Language bindings and also commonly referred to as ECMA 149 (PCTE).

Additional standards applicable to this document are related to the software development environment described in the Institute for Electrical & Electronic Engineers (IEEE) 1209 (Evaluation and Selection of CASE Tools), DoD-STD-1467 (Software Support Environment), NIST-ECMA 500-211 (Reference Model for SEE Frameworks), and MIL-HDBK-782 (Software Support Environment Acquisition).

With respect to software life cycle processes, the MIL-STD-498 *Software Development and Documentation*, and the ISO/IEC 12207 *Software Life Cycle Process* standards are applicable. For the Graphical User Interface (GUI) behavioral design, the MIL-STD-1801 *User Computer Interface* applies.

Standards for APIs are primarily based on Portable Operating Services Interfaces (IEEE POSIX P1003.n series). Additionally API standards created by various industry and consortia are adopted as "Public" APIs. These are X/Open (XPG4 Guide), Open Software Foundation (OSF/DCE), Network Management Forum (NMF) and other expert groups such XAPIA that feed the API standards process into the formal standards developing organizations.

Compliance to the above standards is recommended and the migration to these standards for life cycle development is an ongoing process in this and subsequent DII COE documentation.

3. Windows NT 4.0 Component/Segment Descriptions

The DII COE major component functionally is comprised of: the Kernel, Developer's Toolkit, Infrastructure Services, and Common Support Applications.

3.1 DII COE Kernel

The DII COE Kernel for Microsoft (MS) Windows NT 4.0 provides the core services for the Intel compatible Personal Computer (PC). It is the minimum set of software required for the operation of any DII COE compliant workstation for NT 4.0 regardless of how the workstation is used. The DII COE NT Kernel consist of three components:

- Operating System,
- System/Security Administration, and
- COE Runtime Tools.

The Operating System provides the Graphical User Interface (GUI) and the programmatic access via the application program interface (API) to all DII COE kernel services such as the desktop, systems administration and management, the security, and the COE Runtime Tools. The kernel includes the NT "native" services for the windows operating system capabilities of:

- Windowing,
- System Administration,
- Security, and
- Printing.

There are additional services documented in Windows NT that are not described in the DII COE documentation. The native Windows NT services are described in the NT 4.0 documentation and *NT 4.0 GUI User Guide* available from Microsoft vendor or other suppliers.

The NT is based on Microsoft's Windows Open Services Architecture (WOSA) which provides a single, open-ended interface to the enterprise computing environments. NT hides programming complexities from users and application developers by providing high-level interfaces such as:

- Open Database Connectivity (ODBC),
- Mail Application Program Interface (MAPI),
- Link Services APIs (LSAPI),
- System Networking Architecture APIs (SNA API),
- Windows Sockets, and
- Remote Procedure Calls (RPC).

The NT supports the Component Object Model (COM), Distributed COM (DCOM) and Object Linking and Embedding (OLE) 2.0 specifications that provide various ways to integrate

application components, including features such as visual editing, 'Drag and Drop' between applications, OLE Automation, and structured storage for objects. NT provides both GUI and programmatic access to its underlying features. GUI access is provided with a built-in windowing capability similar to Microsoft Windows 3.1. Programmatic access is provided by the Win32 API.

The Microsoft Windows NT operating system delivered with the COE consists of the basic, vendor-supplied operating system and operating system patches required to operate the COE Runtime, components and segments.

3.1.1 Operating System

Description: MS Windows NT is a POSIX-compliant operating system with a host of built-in features comparable with UNIX based DII COE compliant services provided in the COE Kernel. These services include the following:

- Windows GUI,
- networking and workgroup support,
- administration tools,
- security,
- file sharing and print sharing capabilities,
- data protection,
- Remote Access Service (RAS),
- file backup, and
- command prompt.

Purpose: Windows NT provides basic and advanced operating system features in a single system. Allows a server to provide multiple services for various purposes, e.g., communications server, application server, database server, and file server.

Usage: The NT operating system can be accessed via the built-in GUI or via the command prompt. There is a common set of POSIX calls which are listed in Appendix D. These calls enable the programmer to:

- create, execute, and terminate processes; communicate between processes; and coordinate process operations,
- create, delete and process files and directories,
- activate a process on a timed basis or in response to an external event,
- manage configurable system parameters, and
- access basic security functions inherent in the operating system.

References: Refer to vendor documentation for Windows NT 4.0.
Publications for NT 4.0 are available in bookstores.
Multiple Internet Websites for acquiring Information:
http://www.microsoft.com/
http://moli.microsoft.com/
http://windows-nt.com/
http://www.bhs.com/

3.1.2 Windowing GUI

Description: The NT Program Manager GUI shell is a windows based environment similar to the GUI in Windows 3.1.

Purpose: Provides GUI access to all system functions and windows applications.

Usage: The Program Manager is automatically run when you boot NT and is the GUI shell which is accessed with either mouse or keyboard input. The Windows NT desktop environment provides various services that are equivalent to the DII COE Common Desktop Environments (CDE) services. These NT services consist of:

- Work space management,
- Text editing,
- Terminal emulation,
- Calculator function,
- File management,
- Printer management,
- Style management,
- Session management,
- Login management,
- Process communications, and
- Application integration.

References: Refer to vendor documentation for Windows NT 4.0.
Multiple Internet Websites for acquiring Information:
http://moli.microsoft.com/
http://windows-nt.com/
http://moli.microsoft.com/
http://windows-nt.com/

3.1.3 System/Security Management

The DII COE Security/System Administration (SA) on the NT consists of a single tool. Detailed information about the tool is provided with the DII COE documentation package. The SA consists of the following components:

- Windows NT Client/Server Administration, and
- Security Tools.

The Security/System Administrator is a menu driven GUI application that performs common system management functions performed by a system administrator. The application contains the security management menu interface. The menu interface allows the system administrator to:

- Select printers, configure printers, and manage print jobs,
- Shutdown or restart the system, mount file systems, and initialize hard or floppy disks,
- Load or install COE compliant segments,
- Configure networking parameters on the system.

All the functions for DII COE for Systems/Security Management are facilitated by an integrated Windows NT Administration.

3.1.3.1 Windows NT Client/Server Administration

Description: The DII COE Windows NT 4.0 provides built-in client/server support and comprehensive system and security management. These services are available in the NT system administration tools and are applicable to all NT system features. Some NT programs are multipurpose, i.e., they may act as both a system utility and a system administration tool. For instance, the File Manager is used to copy files as well as to administer the sharing of system storage resources. System administration tools include File Manager, Control Panel, Print Manager, NT Setup, PIF Editor and all tools in the Administrative Tools program group and the Network Administration program group.

Purpose: The NT 4.0 system administration tools are used to configure and secure the NT operating system including establishing users and their privileges. That allows client/server to be established based on access privileges.

Usage: System administration tools are accessed via the GUI and are used to manage the PC workstation.

References: Refer to vendor documentation for Windows NT 4.0.
Multiple Internet Web sites for acquiring Information:
http://moli.microsoft.com/
http://windows-nt.com/

3.1.3.2 Security Tools

Description: NT provides built-in security features equivalent to the security tools used in UNIX. Some NT programs are multipurpose, i.e., they may act as both a system utility and a security administration tool. For instance, the File Manager is used to copy files as well as establishing permissions to protect storage resources. Security administration tools include:

- File Manager,
- Print Manager,
- User Manager for Domains, and
- User Profile Editor.

Purpose: The NT security administration tools are used to configure security settings of users logins.

Usage: The NT security administration tools used by the system administrator to provide access to the system resources. The administrator accesses the tools using the GUI in Windows.

References: Refer to vendor documentation for Windows NT 4.0.
Multiple Internet Websites for acquiring Information:
http://moli.microsoft.com/
http://windows-nt.com/

3.1.4 Print Services

Description: The DII COE Print Services for Windows NT are provided by the Windows built-in NT print manager.

Purpose: To connect, add, modify, configure and share printers. To control print queues and the printing of documents and data.

Usage: The user may control all print jobs or print queues via the Print Manager which is accessible via the Windows GUI.

References: Refer to vendor documentation for Windows NT 4.0.
Multiple Internet Websites for acquiring Information:

3.1.5 COE Runtime Tools

Description: The Runtime tools consist of the 'COEInstaller' (Version 1.0.0.11) for the NT platform. The COE Runtime Tools was developed to provide COE users a method for installing COE segments in the NT environment.

Purpose: The COE Runtime Tools are used to support the segment installation process. DII COE segments were primarily designed to support the system administrator, the applications developer must use these basic tools in constructing any scripts run prior to or after the actual installation.

Usage: COE Runtime Tools are normally executed on the command line or as a menu item in the system administrator application. Some of the COE Runtime Tools do provide functionality through APIs which are of general use. They allow an application to:

- solicit a yes/no type response from the user,
- display a window with a general message or a termination message,
- prompt the user for a text response, and
- prompt for passwords with a verify option.

References: Refer to :
DII COE I&RTS Version 3.0 - Appendix C.
DII COE Application Programmer Interface Reference Guide for Windows NT. Current versions of DII COE NT 4.0 Documents.
Vendor documentation for Windows NT 4.0.
Multiple Internet Websites for acquiring Information:
http://moli.microsoft.com/
http://windows-nt.com/

3.2 Consolidated Developer's Toolkit

The DII COE Developer's Toolkit for NT helps programmers with the creation, installation, testing and management DII COE developed segments. The consolidated developer's toolkit is bundled with the DII COE Kernel and consists of the COE Developer's Toolkit.

3.2.1 COE Developer's Toolkit

Description: The DII COE Developer's Toolkit provides the facilities for programmers to create, install and test DII COE segments.

Purpose: The toolkit aids in the development of COE segments and standardizes the procedures for loading the segments.

Usage: The Developer's Toolkit are used by programmers and software integrators to:

- compute space required by a segment,
- test a segment,
- write segments to an installation medium or package segments,
- temporarily install a segment,
- remove a segment,
- timestamp a descriptor,
- validate a segment, and
- update segment data in the descriptor.

References: Refer to:
DII COE Programmer' s Manual,
DII COE I&RTS,
DII COE Programmer's Guide,
DII COE Version Description Document (VDD), and
DII COE Developer' s Toolkit (Windows NT).

3.3 COE Infrastructure Services

The DII COE Infrastructure services are COE Tools designed for the NT 4.0. These tools are provided as part of the DII COE package and consist of the following components:

- Management Services,
- Presentation and Webs Services,
- Communications Services,
- Data Management Services, and
- Distributed Computing & Obect Management

Some of these services are provided by Windows NT while others are DII COE segments. The DII COE components and segments are loaded using the DII COE install tools and procedures. Any similar or equivalent native NT tools that exist are not components of the DII COE and may not be appropriate for creating DII COE compliant segments.

3.3.1 Management Services

The DII COE Management Services incorporates systems and network management. It provides local as well as remote systems and network administration. These services are provided by the NT and consist of software products that are compatible to the DII COE UNIX components and segments. The following management services are provided in the NT.

- HP NetMetrix
- HP OpenView

The Management Services also includes remote segment installation, remote storage and volume manager, file transfer services and miscellaneous services related to the administration of systems and their respective services.

3.3.1.1 HP NetMetrix

Description: Hewlett Packard NetMetrix (Version 4.70-002) is a UNIX based Network Administration utility to remotely manage the client/server environment.

Purpose: NetMetrix implements a remote network MIB. NetMetrix provides statistics on:

- Network faults and problems,
- Network performance and throughput for capacity planning,
- Network accesses, and
- Network availability.

Usage: NetMetrix is used by network administrators to view a client/server environment and monitor performance.

References: Refer to vendor documentation for HP OpenView 4.11 for Windows NT. Limited information can also be found at HP's homepage; http://www.tmo.hp.com/tmo/ntd

3.3.1.2 HP OpenView 4.11

Description: The HP OpenView product is a UNIX based Network/systems management product that monitors various networked systems, applications, peripherals and networked devices collaboratively with other management systems.

Purpose: HP OpenView provides an enterprise view of the health of a network while remotely managing various systems, applications and peripherals. The OpenView components include;

- IT/Operations
- MeasureWare
- IT/Administration
- OmniBack II

Not all the component functions listed above are used in DII COE.

Usage: HP OpenView is a network/system management tool for DII. It provides management and control capabilities such as auto-discovery, mapping, polling, and alarm/trap status. OpenView is designed to work in a heterogenous client/server enterprise wide system and is capable of managing multiple domains while interoperating in a diverse internetworking environment.

References: Refer to Vendor Documentation for HP OpenView 4.11 for Windows NT and to HP Online Web Site documentation at: http://www.hp.com.

3.3.2 Presentation and Web Services

The DII COE Presentation and Web Services provide a variety of server tools to create and offer information services to various COE clients. Presentation and Web services are part of the DII COE infrastructure and consist of the following list of segments:

- Practical Extraction and Report Language (Perl)
- Browser Services
- News Services
- Multi-Media Presentation

The above COE Services are not available at this time for the Windows NT 4.0. This section will be augmented with video and multimedia presentation services in future releases as these products become available.

3.3.2.1 Browser Services

Description: Netscape Navigator (Version 2.01) provides access to Internet World Wide Web sites. It is a COTS product providing a graphical client browser application for accessing the Web pages and other elements using the Hypertext Transport Protocol (HTTP) over an Internet connection.

Purpose: Netscape Navigator provides graphical tools for accessing the WWW and searching, reviewing, and retrieving information from available sources, which include government sites, commercial sites, and educational sites, to name a few. It also supports electronic mail and access to news groups.

Usage: Netscape Navigator is used to access WWW sites on the SIPRNet using Netscape Navigator capabilities. This allows access to the SIPRNet Web, SIPRNet Newsgroups, and SIPRNet e-mail. Installation and usage of DII COE Browser require licenses for Netscape Navigator version 2.0.1 or above.

References: Refer to Netscape Communications Corp. and to

Netscape on-line documentation (http:/www.netscape.com)
DII System Administrator's Manual

3.3.3 Communications Services

The DII COE Communications infrastructure services provide the capability to interact with other DoD systems, processes and people by exchanging messages and data from the connected systems. The segments for communications are:

- Open Communications Services
- Internet Relay Chatter Client (IRCC)

The DII COE Communications services is provided by Windows NT basic communications support using multi-protocol support to various COE Applications. The NT communications services for COE are limited to the native NT services and to the Internet COE tools provided.

Other COE Communication Services segments are not available at this time for NT 4.0.

3.3.3.1 Open Communications Services

Description: The NT may be configurated for communication with COE servers using the available communications software and utilities of the NT operating system. These are compatible to the DII COE common communications infrastructure when using TCP/IP protocols.

Purpose: The NT communications services support both DCE and non-DCE network environments. This enables NT clients with TCP/IP to send/receive messages.

Usage: Communications software is used to interface applications with other DII COE applications on DoD communications networks.

References: Abstract available on DISA Web site.

3.3.3.2 Internet Relay Chatter Client (IRCC)

Description: IRCC provides the client functionality of chatter style conferencing capabilities. IRCC provides the client programs and files required for multiple users to conduct real-time interactive conferencing on the Internet by connecting to Internet Relay Chatter Server (IRCS).

Purpose: Allows users to communicate with each in conferencing mode while using real-time exchange of messages on an Internet connection.

Usage: Allows users to conference over Internet while exchanging message with each other in real-time over an Internet connection.

References: Refer to online help and/or documentation on the Internet. Use common bulletin boards on freeware or search engines such as Yahoo, Alta Vista, etc.

3.3.4 Data Management Services

The DII COE Data management are COTS products that provide data management services to COE applications development. The major database management for NT is:

- MS ACCESS

The data management services include the high-speed access using standard query language (SQL) to obtain and manipulate various types of information such as: character strings, numbers, money, and date/times. In addition, the data management system facilitates the exchange of data between inter-connected computers over networks. The NT based MS ACCESS provides also an application program interface called ODBC for purposes of accessing data across various other databases.

3.3.4.1 MS ACCESS

Description: Microsoft Access for Windows NT 4.0 is a multi-threaded database server designed to exploit the capabilities of Intel PC based systems and compatible uniprocessor architectures with the capability to deliver a high rate of database scalability, manageability, and performance.

Purpose: MS Access is a COTS database for Windows NT server that provides data management services to various PC based applications.

Usage: MS Access provides the means for application programs to store and retrieve information. It also provides operational support to systems and database administrator functions.

References: Refer to vendor’s product manuals, the supplier's Web page and third-party technical books. Vendor’s Web Site: http://www.microsoft.com.

Other COE Data Management Services (Databases) are not available at this time for Windows NT 4.0.

3.3.5 Distributed Computing & Object Management

The Distributed Computing and Object Management consist of tools for development support using the Distributed Computing Environment (DCE) client and server environment. The Object

Management for the NT server components will be made available in a future release. The DCE client component is generally available on the COE Kernel but listed together herein to facilitate the development of DCE applications supported in DII COE. The NT DCE server component is a third party product with almost all of the DCE Server capabilities. The NT segments for DCE are:

- DCE Client and
- DCE Server.

The Windows NT version of the DII COE release has limited capabilities within the DCE Kernel and does not use services extensions such as DCE Cell Management or Distributed File Services (DFS). Future releases of the NT version of DCE may provide extended services.

3.3.5.1 Distributed Computing Services - Transarc DCE Client

Description: The Transarc DCE Client (version 1.03) is an open systems solution to distributed computing requirements. The DCE Client is composed of a series of services. The NT platform supports only DCE client services.

Purpose: Transarc DCE provides the client communications environment to hide the complexity of network communications from the application developer. This is accomplished by placing layers of services on top of the network services. The DCE Core services support:

- Programs performing many actions concurrently (Threads),
- Direct calls to procedures on remote systems (Remote Procedure Calls or RPC),
- Logically naming objects within a DCE cell (Cell Directory Service or CDS) or throughout a network (Global Directory Service or GDS), and
- Authentication of users and determination of privileges across machines.

Usage: The DCE Runtime simplifies the process of developing client based distributed applications. The DCE Threads are based on the POSIX threading standard and do not introduce any new complexity into the process. RPC calls eliminate explicit programming of network communications and mask differences in data representation across platforms. The directory services allow resources to be identified by name only, without reference to physical location.

References: Refer to vendor documentation from Transarc and/or on-line information on the Internet:
http://www.transarc.com (Transarc's homepage)
http://www.osf.org/dce (Open Systems Foundation)

3.4 DII COE Common Support Applications

The DII COE Common Support Applications are provided to extend Kernel functionality to the user. These applications consist of tools and utilities that are used directly by the users, or a suite of tools which form an actual end-user application system. The Common Support Applications for the NT are:

- Office Automation,
- MCG&I Services,
- Messaging Services,
- Correlations and Situation Display Services.

Elements of the above components may be used by developers for new battlefield management and tactical oriented applications for the warfighter. Windows NT provides the essential Office Automation Services at this time.

3.4.1 Office Automation Services

The NT has limited support for DII COE specific segments. However, the Office Automation services provide a suite of segments popular to all DoD users and developers. These are:

- MS Office for Windows NT,
- Exchange,
- PowerPoint,
- Word, and
- Excel.

Except for the IRC segment that is DII COE all other segments are native mode NT products. The office automation services for NT consist of the Internet Relay Chat for Client and the native Windows NT Office functions of: Exchange, PowerPoint, Word for Windows, and Excel.

3.4.1.1 MS Office Suite for Windows NT

Description: The MS OFFICE SUITE (Version 4.2c) is a DII COE component that provides the 'tool bar' for invoking various office functions such as e-mail, word processing, spreadsheet and presentation graphics. MS Office may be configured as a standalone or as a multi-user server.

Purpose: To provide common office automation tools by presenting a customizable toolbar to the DII COE users and programmers.

Usage: To invoke various applications represented on a push-button tool bar and to provide a high level of integration for information creation, storage and transmission.

References: Refer to vendor documentation for MS OFFICE Suite.
Multiple Internet Websites for acquiring information:
http://moli.microsoft.com/
http://windows-nt.com/

3.4.1.2 Exchange

Description: Microsoft Exchange is a the distributed electronic mail capability for Windows based platforms. It provides gateways to standards based private or public e-mail including SMTP and X.400.

Purpose: Electronic Mail Exchange within a LAN, inter-LANs or across multiple network using WAN.

Usage: Provides functionality to create interpersonal messages with various file attachments for transfer across organizations.

References: Refer to vendor documentation for MS OFFICE Suite or MS Exchange and to Multiple Internet websites for acquiring information:
http://moli.microsoft.com/
http://windows-nt.com/

3.4.1.3 PowerPoint

Description: Microsoft PowerPoint is a Presentation Graphics application program for creation of interactive presentation graphics.

Purpose: Develop presentation content for various public or business functions. PowerPoint creates a variety of presentations for users and is customizable to rehearsed presentations.

Usage: PowerPoint is used to create, edit and present various business material to an audience. It is accessed via the Windows GUI.

References: Refer to vendor documentation for MS OFFICE Suite or MS PowerPoint and to Multiple Internet Websites for acquiring Information:
http://moli.microsoft.com/
http://windows-nt.com/

3.4.1.4 Word (for Windows)

Description: Microsoft Word for Windows is a word processing application program for the creation, editing and printing of information.

Purpose: Write, edit, print and mail various documents and forms using the Windows GUI based editor.

Usage: For various office professionals who create, edit and print documents, letters, forms, etc. Word for Windows is accessed via the Windows GUI.

References: Refer to vendor documentation for OFFICE Suite or Word for Window and to Multiple Internet websites for acquiring information:
http://moli.microsoft.com/
http://windows-nt.com/

3.4.1.5 Excel

Description: Microsoft Excel is a diverse spreadsheet program for statistical and analytical compilation of information.

Purpose: To store and manipulate tabular data, analyze the data, and produce output from the spreadsheet data, e.g., textual displays of totals, automatic calculation of formulae cells or to display resulting data into charts and graphs.

Usage: Excel is accessed via the Windows GUI and is used for statistical analysis and tabulation and manipulation of data.

References: Refer to vendor documentation for MS OFFICE Suite or MS Excel. Multiple Internet Websites for acquiring Information:
http://moli.microsoft.com/
http://windows-nt.com/

3.4.2 Mapping, Charting, Geodesy & Imagery (MCG&I) Services

MCG&I services consist of multiple services accessed by an applications program interface known as Joint Mapping Toolkit (JMTK). JMTK is used for information mapping, planning and other tactical and weather displays for the Warfighter. JMTK is not available for the NT at this time.

3.4.3 Messaging Services

The Messaging Services for tactical communications consist of a set of Common Message Processor (CMP) used for the communications of tactical information. The CMP is not available on the NT at this time.

3.4.4 Correlation Situation Display Services (Unified Build)

The Correlation Situation Display Services consist of various services that are commonly known as Unified Build (UB) and used for tactical situation display for the warfighter. UB is not available on the NT at this time.

Appendix A: Acronym List

ANSI	American National Standards Institute
AOI	Area of Interest
API	Application Programming Interface
APP	Application Portability Profile
ASE	Application Service Entities
C2	Command and Control
C4I	Command, Control Communications, Computers and Intelligence
CASE	Computer-Aided Software Engineering
CDE	Common Desktop Environment
CDS	Cell Directory Service
CMP	Common Message Processor
COE	Common Operating Environment
COM	Component Object Model
COTS	Commercial-Off-The-Shelf
CS	Communications Software
CSE	Client Server Environment
CSTC	Computer Security Technology Center
DBA	Database Administrator
DCE	Distributed Computing Environment
DCOM	Distributed COM
DFS	Distributed File System
DII	Defense Information Infrastructure
DISA	Defense Information Services Agency
DLL	Dynamic Link Libraries
DMA	Defense Mapping Agency
DoD	Department of Defense
DODIIS	DoD Intelligence Information System
DOS	Disk Operating System
DS	Data Store
ECMA	European Computer Manufacturers' Association
E-mail	Electronic Mail
FAQ	Frequently Asked Questions
FIPS	Federal Information Processing Standard
FTP	File Transfer Protocol
GOTS	Government-Off-The-Shelf
GUI	Graphical User Interface
GUNZIP	Name of a decompression tool
GZIP	Name of a Government ZIP Tool (Compression Tool)

HDBK	Handbook
HP	Hewlett Packard
HTML	HyperText Markup Language
HTTP	HyperText Transport Protocol
I&RTS	*DII COE Integration and Runtime Specifications*
ICM	Input Communications Manager
IEC	International Electrotechnical Commission
IP	Internet Protocol
IRCC	Internet Relay Chatter Client
IRCS	Internet Relay Chatter Server
IEEE	Institute for Electrical & Electronic Engineers
ISEE	Integrated Software Engineering Environment
ISO	International Standards Organization
JMTK	Joint Mapping Tool Kits
JOTS	Joint Operational Tactical System
JTA	Joint Technical Architecture
LAN	Local Area Network
LSAPI	Link Services Application Program Interface
LZW	Limpei-Ziv Welch (compression algorithm)
MAPI	Mail Application Program Interface
MCG&I	Mapping, Charting, Geodesy, and Imagery
MIB	Management Information Base
MIL-HDBK	Military Handbook
MS	Microsoft (trademark)
MSVCS	Mail Services
NIS	Network Information Service
NIST	National Institute of Standards and Technology
NMF	Network Management Forum
NT	New Technology (MS-Windows)
ODBC	Open Database Connectivity
OLE	Object Linking and Embeding
OLTP	Online Transaction Processing
OS/2	Operating System/2
OS	Operating System
OSF	Open Systems Foundation, Operational Support Facility
PASC	Portable Application Standards Committee
PC	Personal Computer
PCTE	Portable Common Toolkit Environment

Perl	Practical Extraction and Report Language
PIF	Page Image Format
PIM	Position and Intended Movement
PKZIP	Name of a compression tool
PKZIP	Name of a compression tool
PM	Programmer's Manual
POC	Point of Contact
POSC	Portable Operating System Committee
POSIX	Portable Open System Interface for UNIX
PRM	Programmer's Reference Manual
RAS	Remote Access Service
RPC	Remote Procedure Calls
SA	Security/System Administration
SDE	Software Development Environment
SIPRNet	Secure Internet Protocol Router Network
SeComp	Security Compliance
SIPRNet	Secure Internet Protocol Router Network
SNA	System Network Architecture
SMP	Symmetric Multi-Processor
SMTP	Simple Mail Transfer Protocol
SPI	Security Profile Inspector
SQL	Standard Query Language
SRS	Software Requirements Specification
SS	System Security
STD	Standard (use for DoD documentation)
SUM	Software User’s Manual
TAFIM	Technical Architecture Framework for Information Management
TBD	To Be Determined
TCIM	Tactical Communications Interface Module
TCL	Tool Command Language
TCP	Transmission Control Protocol
Tk	Toolkit
TLC	Tool Command Language
TP	Transaction Processing
UB	Unified Build
UIL	User Interface Language
UNIX	Uniplex Information and Computing System
USMTF	United States Message Text Format
UX	UNIX
VDD	Version Description Document

WAN	Wide Area Network
WEBSv	Netsite Web Server
WIN	Windows
WOSA	Window Open Services Architecture
WWW	World Wide Web
X	X-Windows System
X11R5	X-Windows Version 11, Release 5
XAPIA	X.400 API Application
XM	MOTIF Toolkit (also called ‘MOTIF widgets’)
XPG4	X/Open Programming Guide version 4
XT	X-Window Toolkit Intrinsics

Appendix B: List of COE Tools and APIs (Windows NT 4.0)

TOOL/API NAME	TYPE	Description
DII COE Toolkit - Ref. DII COE Version Description Document for the Kernel (Windows NT) Final Version 3.0.0.3		
COEInstaller 1.0.0.8	Menu Command	Supports the segment installation process. Displays a list of segments that may be installed from floppy diskettes.
DII COE Developer's Toolkit - Ref. DII COE Version Description Document for the Developer's Toolkit (Windows NT 4.0) Version 3.0.0.6		
CalcSpace 1.0.0.6	Menu Command	Computes the space required for the segment specified and updates the hardware descriptor.
CanInstall 1.0.0.8	Menu Command	Tests a segment to see if it can be installed, whcih means that all required segments must already be on the disk and the disk cannot have nay conflicting segments.
MakeInstall 1.0.1.7	Menu Command	Writes one or more segments to an installation medium or packages the segments for distribution over the network..
TestInstall 1.0.0.9	Menu Command	Temporarily installs a segment that already resides on a disk.
TestRemove 1.0.0.8	Menu Command	Removes a segment that was installed by TestInstall.
TimeStamp 1.0.0.8	Menu Command	Puts the current time and date into the `VERSION` descriptor.
VerifySeg 1.0.0.9	Menu Command	Validates that a segment conforms to the rules for defining a segment.
VerUpdate 1.0.1.7	Menu Command	Updates the segment version number, date, and time in the `VERSION` descriptor.
DII COE Toolkit - Ref. DII COE Application Programmer Interface (API) Reference Guide (Windows NT 4.0) Final Version 3.0.0.6		
COEAskUser	Code	Creates an interface that prompts the user for a question and gives a choice of `Yes` or `No` buttons to select. The message question and the answer button labels can be assigned by the user.
COEFindSeg	Code	Searches a given segment directory for a given segment name to see if it is on disk and can be read. Returns the segment directory, name, prefix, type and attribute.
COEInstError	Code	Creates an interface that displays an installation error message.
COEMsg	Code	Creates an interface that displays a message.
COEPrompt	Code	Displays a prompt and a text area to accept user input.
COEPromptPasswd	Code	Displays an optional `Password` prompt message and an option `Verify` prompt.

Appendix C: List of POSIX Function Calls

Process Control

_exit() - terminate a process after flushing any pending output
execle() - provides an interface to the execve call with know arguments
execve() - transfer to entry point of new executable - vector of arguments
fork() - create a new process
getpgrp() - get process group of specified process
getpid() - get process identification
kill() - send signal to a process
pause() - stop until signal
pipe() - create an interprocess channel
setpgid() - join or create new process group
setsid() - create session and set process group ID
sleep() - suspend execution for interval
wait() - wait for process to terminate
waitpid() - wait for specific child process
tcgetpgrp() - get foreground process group ID
tcsetpgrp() - set foreground process group ID
times() - get process times
uname() - get name of current system

File Operations

access() - determines the accessibility of a file
chdir() - change working directory
chmod() - change mode of file
chown() - change owner and group of a file
close() - delete a file descriptor
creat() - create a new file
dup2() - duplicate an open file descriptor, returning a new descriptor
dup() - duplicate an open file descriptor
fcntl() - test, set or clear read or write locked file
fstat() - get file status
link() - link to a file
lseek() - move read or write pointer associated with a file or open device
mkdir() - make a directory file
mkfifo() - make a FIFO special file
open() - open file for read or write and return descriptor
pathconf() - get configurable pathname variables

read() - read from a file
rename() - change the name of a file
rmdir() - remove a directory file
stat() - get file status
tcflush() - discard data written but not transmitted or received but not read
tcdrain() - wait until all output written to the *fildes* object transmitted
tcflow() - suspend transmission or reception of dataon the *fildes* object
tcgetattr() - get parameters associated with *fildes* - store in *termios* structure
umask() - set file creation mask
unlink() - remove directory entry
ustat() - get file system statistics
utime() - set file times
write() - write on a file

Signal Processing

alarm() - schedule signal after specified time
sigaction() - software signal facilities
sigaddset() - add specified signal from specified signal set
sigdelset() - delete specified signal from specified signal set
sigemptyset() - initializes signal set so that all signals are excluded
sigfillset() - initializes signal set so that all signals are included
sigismember() - tests whether signal is member of signal set
sigpending() - examine pending signals
sigprocmask() - examine and change blocked signals
sigsuspend() - wait for signal

System Parameters

cfgetispeed() - returns the input baud rate stored in the *termios* structure
cfgetospeed() - returns the output baud rate stored in the *termios* structure
cfsetispeed() - sets the input baud rate stored in the *termios* structure
cfsetospeed() - sets the output baud rate stored in the *termios* structure
sysconf() - get configurable system variables
tcsendbreak() - cause transmission of a stream of zero valued bits
tcsetattr() - set parameters associated with terminal from *termios* structure
time() - get date and time

Group and User IDs

getegid() - get effective group ID
geteuid() - get effective user ID

getgid() - get real group ID
getgroups() - get group access list
getuid() - get user identity
setgid() - set real and effective group ID
setuid() - set user and group ID

Appendix D: Application Programmer Interface (API) Reference Guide for Windows NT 4.0

Preface

The following conventions have been used in this document:

[HELVETICA FONT]	Used to indicate keys to be pressed. For example, press [RETURN].
`Courier Font`	Used to indicate entries to be typed at the keyboard, Windows NT commands, titles of windows and dialog boxes, file and directory names, and screen text. For example, execute the following command: `A:\setup.exe`
"Quotation Marks"	Used to indicate prompts and messages that appear on the screen.
Italics	Used for emphasis.

D.1 Introduction

D.1.1 Overview

This document provides information and guidance needed for using Defense Information Infrastructure (DII) Common Operating Environment (COE) Version 3.0.0.6 public Application Programmer Interfaces (APIs) for Windows NT 4.0. This document contains manual pages for all of the public APIs for the DII COE toolkit for Windows NT 4.0.

This guide is divided into the following two sections and one appendix:

Section	Page
Introduction Provides an overview of the *DII COE API Reference Guide* and lists documents that can be referenced for further information about the DII COE APIs.	32

Calling the DII COE Tools Using the API Toolkit
Includes manual pages for the following DII COE tools:
`COEAskUser` `COEFindSeg`
`COEInstError` `COEMsg`
`COEPrompt` `COEPromptPasswd`
	33

Notes Provides additional information about the `COEPromptPasswd` tool.	53

Each manual page includes a synopsis, parameters, a description, return values, notes, a reference to related functions, and an example.

Descriptions assume familiarity with the C programming language and with the DII COE development environment.

D.1.2 Additional Sources of Information

Reference the following documents for more information about the APIs for the DII COE toolkit:

- *Defense Information Infrastructure (DII) Common Operating Environment (COE) Integration and Runtime Specification* Version 2.0, DII COE I&RTS:Rev 2.0, Inter-National Research Institute, October 23, 1995

- *Defense Information Infrastructure (DII) Common Operating Environment (COE) Programming Guide (Windows NT 4.0)* Version 3.0.0.6, DII.3006.NT40.PG-1, Inter-National Research Institute, February 5, 1997.

D.2 Calling the DII COE Tools Using the API Toolkit

The API toolkit provides developers with an interface to the DII COE runtime tools. A program can link with a public API to display and retrieve segment information. This section defines the interface for each call, including return values and necessary parameters. This section provides information about using the following APIs:

- `COEAskUser`
- `COEFindSeg`
- `COEInstError`
- `COEMsg`
- `COEPrompt`
- `COEPromptPasswd`.

NOTE: Reference Attachment A, *Notes*, for more information about using `COEPromptPasswd`.

The format of each manual page is as follows:

NAME
Function Name—Provides a brief description of the function.

SYNOPSIS
Presents the calling syntax for the routine, including the declarations of the arguments and the return type. For example:

```
                returntype XFunctionName (type1 *arg1, type2 *arg2, type3
*arg3);
```

PARAMETERS
Describes each of the parameters used by the function.

DESCRIPTION
Describes what the function does and what events or side effects it causes.

RETURNS
Describes what the function returns.

NOTE
Provides any notes about the function.

SEE ALSO
Provides a reference to related functions.

EXAMPLE
Provides an example of how to use the function.

D.2.1 COEAskUser

NAME
COEAskUser—COEAskUser() displays a window with a question and two answer buttons.

SYNOPSIS

```
#include <DIITools.h>

int COEAskUser
(
                char *question,
                char *b1_label,
                char *b2_label
);
```

PARAMETERS

`char *question`

`question` - Null terminated string. The question to present to the user.

`char *b1_label`

`b1_label` - Null terminated string. The equivalent of the `Yes` button; selection causes the function to return a `1` (`TRUE`).

`char *b2_label`

`b2_label` - Null terminated string. The equivalent of the `No` button; selection causes the function to return a `0` (`FALSE`).

DESCRIPTION

The COEAskUser library function creates an interface that prompts the user for a question and gives a choice of `Yes` or `No` buttons to select, where `Yes` and `No` are the default labels assigned. The message question and the answer button labels can be assigned by the user. Null values for `b1_label` and `b2_label` will display default button labels.

RETURNS

`COEFAILURE`

Failure - The interface could not be displayed. On returning `COEFAILURE`, `COEerrno` is set to:
`COEERR_NO_DISPLAY` - Window could not be displayed.
`COEERR_NO_MESSAGE` - Question string not passed in.

`TRUE`

True - The equivalent `Yes` button selected by the user.

`FALSE`

False - The equivalent `No` button selected by the user.

NOTE

To support multi-line questions, place `\n` in the desired locations in the question string.

SEE ALSO

COEMsg, COEPrompt, and COEPromptPasswd.

EXAMPLE: COEAskUser

```
/*
 To build this routine use the following command(substitute your
 location for the DII_DEV directory, Motif libraries and includes) :

HP:
 cc -Aa -o COEAskUser_example COEAskUser_example.c -I/h/DII_DEV/include
 -I/usr/include/Motif1.2 -I/usr/include/X11R5 -L/h/DII_DEV/libs
 -lCOETools -lCOE -lPrintClient -L/usr/lib/X11R5 -L/usr/lib/Motif1.2 -lXm -lXt
-lX11

SOLARIS:
 cc -o COEAskUser_example COEAskUser_example.c -I/h/DII_DEV/include
 -I/usr/include/Motif1.2 -I/usr/openwin/include -L/h/DII_DEV/libs
 -lCOETools -lCOE -lPrintClient -L/usr/openwin/lib -L/usr/lib/Motif1.2 -lXm -
lXt -lX11 -lgen
```

```
NT:
  In your compile environment, make sure your include file path
  includes DII_DEV/include and your library path includes DII_DEV/libs.
  Link COEUserPrompts.lib during compilation.  (The COEUserPrompts.dll
  will be required during execution.)
*/
#include <stdio.h>
#include <DIITools.h>

/*****************************/
/* COEAskUser_example        */
/*****************************/
int main(int argc, char *argv[])
{
  char         b1_lab[] = "MY_YES";
  char         b2_lab[] = "MY_NO";
  char         message[]="This is my test Message";
  int          ret_val;

  /* Call DII/COE Library Function */
  ret_val = COEAskUser(message, b1_lab, b2_lab);

  exit(ret_val);
}
```

D.2.2 COEFindSeg

NAME

COEFindSeg—COEFindSeg() returns information about a requested segment.

SYNOPSIS

```
#include <DIITools.h>

int COEFindSeg
(
                  int use_installed,
                  char *in_segdir,
                  char *in_segname,
                  char *prefix,
                  char *seg_type,
                  char *seg_attrib,
                  char *actual_dir
);
```

PARAMETERS

`int use_installed`

`use_installed` - Installed segments only - `1`, any segments - `0`. Note: This flag is ignored on Windows NT.

`char *in_segdir`

`in_segdir` - Null terminated string. The full directory path leading up to and including `segdir`.

`char *in_segname`

`in_segname` - Null terminated string. The segment name of the segment residing in `segdir`.

`char *prefix`

`prefix` - Segment prefix returned.

`char *seg_type`

`seg_type` - Segment type returned.

`char *seg_attrib`

`seg_attrib` - Segment attribute returned if available.

`char *actual_dir`

`actual_dir` - Segments actual directory returned.

DESCRIPTION

The COEFindSeg library function searches a given segment directory for a given segment name to see if it is on disk and can be read. If no directory is specified, COEFindSeg searches for installed segments by default. When the segment is found, COEFindSeg returns the segment directory, name, prefix, type, and attribute (if the segment has an attribute).

RETURNS

`COESUCCESS`

Success - The function was successful in finding the requested segment.

`COEFAILURE`

Failure - The function was not successful in finding the requested segment. On returning `COEFAILURE`, `COEerrno` is set to:

`COEERR_SEG_NOT_FND` - Requested segment not found.

`COEERR_NULL_PARAMS` - Both segment name and directory parameters were NULL or empty.

`COEERR_FOUND_NOT_INSTALLED` - Requested segment was not installed as requested.

NOTE

None.

SEE ALSO

COEAskUser, COEInstError, and COEMsg.

EXAMPLE: COEFindSeg

```
/*
 To build this routine use the following command(substitute your
 location for the DII_DEV directory, Motif libraries and includes) :

HP:
 cc -Aa -o COEFindSeg_example COEFindSeg_example.c
 -I/h/DII_DEV/include -I/usr/include/Motif1.2 -I/usr/include/X11R5
 -L/h/DII_DEV/libs -lCOETools -lCOE -lPrintClient -L/usr/lib/X11R5
 -L/usr/lib/Motif1.2 -lXm -lXt -lX11

SOLARIS:
 cc -o COEFindSeg_example COEFindSeg_example.c -I/h/DII_DEV/include
```

```
 -I/usr/include/Motif1.2 -I/usr/openwin/include -L/h/DII_DEV/libs
 -lCOETools -lCOE -lPrintClient -L/usr/openwin/lib -L/usr/lib/Motif1.2 -lXm -
lXt -lX11 -lgen

NT:
  In your compile environment, make sure your include file path
  includes DII_DEV/include and your library path includes DII_DEV/libs.
  Link COECom.lib, COESeg.lib and COETools.lib during compilation.

*/
#include <stdio.h>
#include <DIITools.h>

/*****************************/
/* COEFindSeg_example        */
/*****************************/
int main(int argc, char *argv[])
{
  char segdir[] = "";
  char segname[] = "X Windows";
  int  ret_val;
  char    out_prefix[8];
  char    out_segtype[81];
  char    out_segattr[81];
  char    out_actualdir[257];


  /* Call DII/COE Library Function */
 ret_val = COEFindSeg(1,segdir,
segname,out_prefix,out_segtype,out_segattr,out_actualdir );
 if( ret_val == COESUCCESS )
 {
    printf("Found segment '%s' prefix '%s' type '%s' at
'%s'\n",segname,out_prefix,out_segtype,out_actualdir);
 }
 else
 {
    printf("Segment '%s' not found \n", segname);
 }


  exit(ret_val);
}
```

D.2.3 COEInstError

NAME

COEInstError—COEInstError() displays a window with the error message.

SYNOPSIS

```
#include <DIITools.h>

int COEInstError
(
                char *message
);
```

PARAMETERS

```
char *message
```

`message` - Null terminated string. Error message to display.

DESCRIPTION

The COEInstError library function creates an interface that displays an installation error message. The message can be assigned by the caller of the function. `COEFAILURE` is always returned. It is up to the calling application to perform appropriate cleanup and return a failed exit status to its parent program (if applicable).

RETURNS

`COEFAILURE`

Failure - Always returned to signal the calling process that an error has occurred. On returning `COEFAILURE`, `COEerrno` is set to:

`COEERR_NO_DISPLAY` - Window could not be displayed.

`COEERR_NO_MESSAGE` - Message string not passed in.

NOTE

To support multi-line questions, place `\n` in the desired locations in the message string.

SEE ALSO

COEMsg, COEPrompt, and COEPromptPasswd.

EXAMPLE: COEInstError

```
/*
 To build this routine use the following command(substitute your
 location for the DII_DEV directory, Motif libraries and includes) :

HP:
 cc -Aa -o COEInstError_example COEInstError_example.c -I/h/DII_DEV/include
 -I/usr/include/Motif1.2 -I/usr/include/X11R5 -L/h/DII_DEV/libs
 -lCOETools -lCOE -lPrintClient -L/usr/lib/X11R5 -L/usr/lib/Motif1.2 -lXm -lXt
-lX11

SOLARIS:
 cc -o COEInstError_example COEInstError_example.c -I/h/DII_DEV/include
 -I/usr/include/Motif1.2 -I/usr/openwin/include -L/h/DII_DEV/libs
 -lCOETools -lCOE -lPrintClient -L/usr/openwin/lib -L/usr/lib/Motif1.2 -lXm -
lXt -lX11 -lgen

NT:
  In your compile environment, make sure your include file path
  includes DII_DEV/include and your library path includes DII_DEV/libs.
  Link COEUserPrompts.lib during compilation.  (The COEUserPrompts.dll
  will be required during execution.)
*/
#include <stdio.h>
#include <DIITools.h>

/*****************************/
/* COEInstError_example      */
/*****************************/
int main(int argc, char *argv[])
{
  char         message[]="Cannot Access ABC Server";
  int          ret_val;
```

```
  /* Call DII/COE Library Function */
  ret_val = COEInstError(message);

  exit(ret_val);
}
```

D.2.4 COEMsg

NAME
COEMsg—COEMsg() displays a window with a message of choice.

SYNOPSIS
```
#include <DIITools.h>

int COEMsg
(
                char *message
);
```

PARAMETERS
`char *message`

`message` - Null terminated string. Message to display to the user.

DESCRIPTION
The COEMsg library function creates an interface that displays a message. The message can be assigned by the caller of the function, which makes it versatile enough to be used as a message for any program.

RETURNS
`COESUCCESS`

Success - The function was successful in displaying the message.

`COEFAILURE`

Failure - The function was not successful in displaying the message. On returning `COEFAILURE`, `COEerrno` is set to:
`COEERR_NO_DISPLAY` - Window could not be displayed.
`COEERR_NO_MESSAGE` - Question string not passed in.

NOTE
To support multi-line questions, place `\n` in the desired locations in the message string.

SEE ALSO
COEAskUser, COEInstError, COEPrompt, and COEPromptPasswd.

EXAMPLE: COEMsg

```
/*
 To build this routine use the following command(substitute your
 location for the DII_DEV directory, Motif libraries and includes) :
```

```
HP:
 cc -Aa -o COEMsg_example COEMsg_example.c -I/h/DII_DEV/include
 -I/usr/include/Motif1.2 -I/usr/include/X11R5 -L/h/DII_DEV/libs
 -lCOETools -lCOE -lPrintClient -L/usr/lib/X11R5 -L/usr/lib/Motif1.2 -lXm -lXt
-lX11

SOLARIS:
 cc -o COEMsg_example COEMsg_example.c -I/h/DII_DEV/include
 -I/usr/include/Motif1.2 -I/usr/openwin/include -L/h/DII_DEV/libs
 -lCOETools -lCOE -lPrintClient -L/usr/openwin/lib -L/usr/lib/Motif1.2 -lXm -
lXt -lX11 -lgen

NT:
  In your compile environment, make sure your include file path
  includes DII_DEV/include and your library path includes DII_DEV/libs.
  Link COEUserPrompts.lib during compilation.  (The COEUserPrompts.dll
  will be required during execution.)
*/
#include <stdio.h>
#include <DIITools.h>

/*****************************/
/* COEMsg_example            */
/*****************************/
int main(int argc, char *argv[])
{
  char         message[]="XYZ Has Been Updated";
  int          ret_val;

  /* Call DII/COE Library Function */
  ret_val = COEMsg(message);

  exit(ret_val);
}
```

D.2.5 COEPrompt

NAME

COEPrompt—COEPrompt() displays a window with a user-assigned prompt and an editable text area for user input.

SYNOPSIS

```
#include <DIITools.h>

int COEPrompt
(
                  char *prompt,
                  int max_len,
                  char *text_return
);
```

PARAMETERS

`char *prompt`

`prompt` - Null terminated string. Prompt text to display.

`int max_len`

`max_len` - Maximum length of text accepted in the editable text field.

`char *text_return`

`text_return` - Null terminated string. Text returned—must be allocated to `max_len + 1`.

DESCRIPTION

The COEPrompt library function creates an interface that displays a prompt and a text area to accept user input. The prompt can be assigned by the caller of the function. The user input will be sent to `stdout`.

RETURNS

`COESUCCESS`

Success - The function was successful in displaying the prompt and the text area.

`COEFAILURE`

Failure - The function was not successful in displaying the prompt and the text area. On returning `COEFAILURE`, `COEerrno` is set to:

`COEERR_NO_DISPLAY` - Window could not be displayed.

`COEERR_NO_MESSAGE` - Prompt string not passed in.

NOTE

To support multi-line questions, place `\n` in the desired locations in the message string.

SEE ALSO

COEAskUser, COEInstError, COEMsg, and COEPromptPasswd.

EXAMPLE: COEPrompt

```
/*
 To build this routine use the following command(substitute your
 location for the DII_DEV directory, Motif libraries and includes) :

HP:
 cc -Aa -o COEPrompt_example COEPrompt_example.c -I/h/DII_DEV/include
 -I/usr/include/Motif1.2 -I/usr/include/X11R5 -L/h/DII_DEV/libs
 -lCOETools -lCOE -lPrintClient -L/usr/lib/X11R5 -L/usr/lib/Motif1.2 -lXm -lXt
-lX11

SOLARIS:
 cc -o COEPrompt_example COEPrompt_example.c -I/h/DII_DEV/include
 -I/usr/include/Motif1.2 -I/usr/openwin/include -L/h/DII_DEV/libs
 -lCOETools -lCOE -lPrintClient -L/usr/openwin/lib -L/usr/lib/Motif1.2 -lXm
-lXt -lX11 -lgen

NT:
  In your compile environment, make sure your include file path
  includes DII_DEV/include and your library path includes DII_DEV/libs.
  Link COEUserPrompts.lib during compilation.  (The COEUserPrompts.dll
  will be required during execution.)
*/
#include <stdio.h>
#include <DIITools.h>
```

```
/*****************************/
/* COEPrompt_example         */
/*****************************/
int main(int argc, char *argv[])
{
  char         message[]="Please Enter What Server To Access : ";
  int          input_text_len = 15;
  int          ret_val;
  char                    ret_text[16];

  /* Call DII/COE Library Function */
  ret_val = COEPrompt(message, input_text_len, ret_text);

  exit(ret_val);
}
```

D.2.6 COEPromptPasswd

NAME

COEPromptPasswd—COEPromptPasswd() displays a window with an optional `Password` prompt message and an optional `Verify` prompt.

SYNOPSIS

```
#include <DIITools.h>

int COEPromptPasswd
(
                  int max_text_len,
                  int user_wants_verify_prompt,
                  char *prompt,
                  char *passwd_return
);
```

PARAMETERS

`int max_text_len`

`max_text_len` - Maximum characters the user may enter for the password.

`int user_wants_verify_prompt`

`user_wants_verify_prompt` - Flag whether to display a `Verify` prompt.

`char *prompt`

`prompt` - Null terminated string. Optional prompt to display text.

`char *passwd_return`

`passwd_return` - Password returned.

DESCRIPTION

The COEPromptPasswd library function creates an interface that displays an optional `Password` prompt message and an optional `Verify` prompt. The optional `Verify` prompt can either be (1) displayed by passing in `TRUE` or (2) not displayed by passing in `FALSE` through the third parameter. The password entered will be sent to `stdout`.

RETURNS

`COESUCCESS`

Success - The function was successful. (This function is always successful if the `Verify` prompt is not displayed.)

`COEFAILURE`

Failure - The interface cannot be displayed, or the `Password` and `Verify` text are not the same. On returning `COEFAILURE`, `COEerrno` is set to:
`COEERR_NO_DISPLAY` - Window could not be displayed.
`COEERR_EMPTY_FIELD` - No Password text or verify text returned.
`COEERR_NO_MATCH` - Password and Verify string do not match.

NOTE
The password has a default maximum length of 40 characters when `max_text_len` is less than or equal to zero. The maximum password length can be increased or decreased by the caller of the function via the `max_text_len` field. If the input maximum length is exceeded, the input will be truncated.

To support multi-line questions, place `\n` in the desired locations in the message string.

SEE ALSO
COEAskUser, COEInstError, COEMsg, and COEPrompt.

EXAMPLE: COEPromptPasswd

```
/*
 To build this routine use the following command(substitute your
 location for the DII_DEV directory, Motif libraries and includes) :

HP:
 cc -Aa -o COEPromptPasswd_example COEPromptPasswd_example.c
 -I/h/DII_DEV/include -I/usr/include/Motif1.2 -I/usr/include/X11R5
 -L/h/DII_DEV/libs -lCOETools -lCOE -lPrintClient -L/usr/lib/X11R5
 -L/usr/lib/Motif1.2 -lXm -lXt -lX11

SOLARIS:
 cc -o COEPromptPasswd_example COEPromptPasswd_example.c  -I/h/DII_DEV/include
 -I/usr/include/Motif1.2 -I/usr/openwin/include -L/h/DII_DEV/libs
 -lCOETools -lCOE -lPrintClient -L/usr/openwin/lib -L/usr/lib/Motif1.2 -lXm
-lXt -lX11 -lgen

NT:
  In your compile environment, make sure your include file path
  includes DII_DEV/include and your library path includes DII_DEV/libs.
  Link COEUserPrompts.lib during compilation.  (The COEUserPrompts.dll
  will be required during execution.)
*/
#include <stdio.h>
#include <DIITools.h>

/*****************************/
/* COEPromptPasswd_example    */
/*****************************/
int main(int argc, char *argv[])
{
  int          input_text_len = 10;
  int                      ret_val;
  char                     ret_passwd[11];

  /* Call DII/COE Library Function */
  ret_val = COEPromptPasswd(input_text_len, 1,
                  "Please enter passwd for MyUser", ret_passwd);
  if (ret_val == COESUCCESS) {
                  printf("Password Is Correct\n");
  }
  else {
                  printf("Password Is Incorrect\n");
```

```
  }

  exit(ret_val);
}
```

This page intentionally left blank.

Attachment A - Notes

COEPromptPasswd

The password has a default maximum length of 40 characters when `max_text_len` is less than or equal to zero. The maximum password length can be increased or decreased by the caller of the function via the `max_text_len` field. If the input maximum length is exceeded, the input will be truncated.

The software does not check if the `max_text_len` of the password is less than or equal to zero. Therefore, the value will not default to 40.

This page intentionally left blank.